

ECHO ACLs and Roles Ops Concept

Prepared by: Matt Cechini, Mike Pilone, and Lisa Pann

1. Operations Concept

This document describes proposed changes to the access control level (ACL) system to reduce complexity and implement requested functionality which includes expanding the system beyond the current metadata catalog scope.

1.1 Background

The current ACL system feature-set is a carryover from earlier, versions of ECHO. Now that that API is used regularly by providers and client applications, a number of limitations, complexities, and unnecessary features have been identified. In order to address these issues, the ACL API needs to be modified based on past experience and future needs.

At the same time, the need for a more encompassing ACL system has been identified. To support more complex use cases proposed by the DAACs, ECHO needs to support ACLs for more domain objects including but not limited to provider orders and provider policies.

1.2 General Challenges

The general issues are briefly described in this section. For more detail on this issue, refer to NCRs:

- 11003839 An admin with a role for a single provider can not get an order for any other provider.
- 11004240 ValidateOrder does not check user owns the order.
- 11004222 Admin has inconsistent access to other user's orders
- 11003141 Echo:WebPump Attempts to restrict collection granules by temporal range does not work.
- 11004485 Drop the condition name and description from data rule conditions
- 11004171 ECHO - Improve usability of PUMP's Rules page
- 11004015 Setting ordering and viewing permissions in PUMP.
- 11003354 ECHO 10.0 ECHO Roles
- 11003582 ECHO 10.0 ETE test - Temporal ACL not working properly
- 11004655 ECHO 10.13 TS1. Granule restriction is not working on WIST searches
- 11004013 Setting Visibility and Order Options in PUMP
- 11004671 Remove automatic provider context setting when logging in
- 11004007 Application of temporal/rolling temporal data access rules when target fields are null

1.2.1 Difficulty Reviewing Rules in PUMP

Pump currently displays all the rule information on a single page for the provider. Once a couple of rules are created, this page can become unwieldy. (NCR 11004171)

1.2.2 API Inconsistency

The ACL rules are currently defined using granule URs or collection data set IDs. While the rest of ECHO uses the ECHO catalog item GUID. This difference can be confusing for client developers who are used to using the GUIDs.

1.2.3 Inability to Select Multiple Permissions

Currently, ACL rules grant a single permission which are the ability to view or order catalog items. NCR 11004015 requests that for a single set of collections, both order and view permissions can be granted at the same time. This could be done as a client feature in PUMP or a change to ECHO to allow a single rule to apply multiple grants. (NCR 11004015, 11004013)

1.2.4 Confusing Comparator Support

An ACL rule contains a comparison field which can be used to express greater than, less than, equal to, etc. However not all the rules support all the comparison operators, which results in a runtime rejection when a rule is created. Some conditions, such as the Boolean condition, support the equal to and not equal to operators while at the same time taking a Boolean value. This results in multiple ways to create the same rule behavior which can lead to confusion (NCR 11004007).

1.2.5 Inability to Limit ACL to Granules within a Collection

Providers have expressed an interest in being able to have an ACL apply to only granules within a given collection. The current ACL rules can be created to apply to either granules or collections, but not to granules within a collection. The API does not easily support specifying the containing collection because the same field is overloaded for collection and granule IDs. (NCR 11003582, 11003141).

1.2.6 Condition Exclusivity

Due to limitations in some of the older XML tools, the original API Schema did not implement condition elements as a <choice> in the ACL rule, but instead all elements were listed as optional. While the textual documentation explains the fact that only one condition can be present, the schema does not enforce this resulting in a runtime rejection when a rule is created.

1.2.7 Inability to Apply ACLs to Other Domain Objects

ECHO currently only supports ACLs on catalog item metadata. Providers have expressed an interest in creating groups of users with limited permissions on other domain objects such as provider orders and provider policies. One option to support this was the introduction of a new user role; however after some review it was determined that the expansion of general ACLs would better support this requirement and provide for future needs (NCR 11003354, 11004222, 11004240).

1.2.8 Role vs. Group Confusion

ECHO supports both user roles (provider and administrator) as well as dynamic groups. These concepts are extremely similar and can be viewed as the same given the correct implementation of groups (Ferraiolo, Kuhn, & Chandramouli, 2003, p. 54). By supporting both groups and roles, ECHO's security model is more complex and less user friendly (NCR 11003354).

1.2.9 Confusing Provider Context Behavior

If a user has a single provider role, ECHO automatically places that user in the matching provider context when the user logs in. ECHO also ignores many ACLs for users with a provider context when searching and returning metadata to allow providers to see all data (even restricted data). While originally designed as a convenience feature, the combination of these two behaviors can cause confusion to users in clients other than PUMP. For example, if a user with a single provider role logs into WIST, ECHO will automatically place them in the provider context allowing them to search and order data that should not be publicly accessible. While this doesn't pose a security problem, it doesn't allow the user to verify that ACLs created in PUMP are being applied correctly (NCR 11003839, 11004671).

1.3 Proposed Changes

The following concepts are a part of the proposed changes to the ACL API which are being made so that the API will be more client-friendly and flexible. The concepts discussed in this section have been developed to address the issues which were identified in Sections 1.2.

1.3.1 Extension of ACLs Beyond the Metadata Catalog

ECHO will support the creation and administration of ACLs for the metadata catalog as well as specific business domain objects such as provider orders, policies, option definitions, option assignments or holding reports. See Section 3 for further discussion regarding the domain objects which will be controlled by ACLs.

1.3.2 All Data Restricted by Default

Most providers have established restriction rules that restrict all data and then have explicit permission rules that permit access to public data to all and non-public data to specific groups. To simplify this setup and to remove some confusion with regard to the application of the rules, all data in ECHO will be considered restricted unless an ACL explicitly grants permission to a user group (or the general public). This should simplify rule creation by making all rules permission rules.

This concept will also apply to ECHO domain objects making operations such as viewing provider orders or policies restricted unless the user is in a group that has been granted explicit permissions.

1.3.3 Merging of Role and Group Concepts

The concept of a role and a group will be merged into a single concept in ECHO. A user will no longer be granted or revoked roles but will simply be a member of one or more groups. For backward compatibility, ECHO will still return role information in a user profile for provider roles so existing client applications can still identify which providers to which the user can set context. The `revokeRole` and `grantRole` functionality will be removed. The role information in the user's profile may also be removed in the future once client applications are updated.

1.3.4 Removal of Provider Role Data Discovery Exemption

Provider users (users acting on behalf of a provider in a "provider context") will no longer be exempt from ACLs for catalog item metadata. To provide full access to provider metadata, providers should create a group and add all "provider" users to that group. Then ACLs can be constructed as desired to allow the "provider" group to have full access to any metadata.

Currently, if a user has only one provider role, when they log into ECHO (or WIST), ECHO automatically sets that user's context to the provider for which they have provider role. This 'functionality' will be removed. As was mentioned previously, Provider users will have access to data according to the catalog item ACLs that are associated with groups of which they are a member.

1.3.5 Removal of Visible and Orderable Filters

The orderable and visible flags will no longer be checked by ECHO when determining access to a particular item for a user. Removing this extra type of access control reduces the complexity when

configuring security on a catalog item. Providers looking for similar functionality should consider using ACLs on restriction flag in the metadata.

The visibility and orderable flags will remain in the metadata for backward compatibility; however the fields will most likely be removed in the future. ECHO Operations has worked with its Data Partners to move towards a scheme where these fields are not needed.

1.3.6 Multiple Permissions Per ACL

A single ACL can be configured to include multiple permissions for a group such as read and order or read and write. This should reduce the total number of rules needed because in most cases the ACLs for ordering and viewing are the same. The functionality of only permitting a single type of action will continue to be supported.

1.3.7 Extended Conditions for Catalog Item ACLs

Currently most catalog item ACLs are assigned to collections using the data set ID directly. After reviewing many of the current rules, a pattern emerges in which most of the assignments are to groups of collections with similar properties. For example, all the collections with the same short name or data set ID prefix are assigned to the same rules. For example, a provider may have a rule that says “For all the ‘GROUND ICE.*’ collections, apply this rule”. Section 3 of this document provides further discussion regarding the supported conditions and their comparison operators that will be used for catalog item ACLs.

An advantage of this approach is that the number of overall ACLs may be reduced because fewer, pattern based ACL conditions could be created. Also, new collections may automatically be covered by an ACL requiring no further setup after Ingest.

A disadvantage of this approach is that the ACL assignments may become more complicated because the logic of how the conditions are applied which is discussed in detail later. As stated in the advantages, the automatic assignment of ACLs may be considered a disadvantage if a new collection wasn’t supposed to be immediately affected by an ACL but it just happens to match an existing ACL condition based on the name or other properties.

1.3.8 Default Access Changes

Within the new ACL structure and in combination with current API functionality, the following items would require providers to create an ACL explicitly permitting access:

- Option Definitions
- Authenticator Definitions
- Extended Services

We propose that all instances of these items will default to being readable by all ECHO registered and guest users. This will eliminate the provider’s need to create ACLs to manage access to these three resources.

1.3.9 ECHO Operations Catalog Item Access

It is proposed that ECHO Operators will be able to assign themselves permissions to view and order all data for all data providers. Data Partners will not need to grant access to ECHO Operations or WIST Valids, as they currently do. ECHO Operations will work closely with Data Partners if orders are to be placed for testing purposes.

1.4 Data Partner Impact

The following sections describe how the proposed functionality in this document will affect data partners. It is important that data providers utilize the Testbed and/or Partner Test systems in order to verify that there are no unintended consequences associated with this approach.

1.4.1 Metadata Exports

There are no changes to how ECHO processes metadata exports, however the values for the visible and orderable elements will no longer be used by ECHO for data management.

1.4.2 ACL Migration

ECHO Operations will work with each Data Partner to develop a tailored ACL structure for catalog item access, data management actions, and user/order management actions. This plan will be implemented in the Partner Test system where providers may verify that the intended access control mechanisms function properly. When this capability is released into ECHO Operations, there will need to be a transition from the old ACL structure to the new roles. ECHO Operations and development will develop a migration strategy to minimize the amount of time where a provider's data will not be visible. This activity will happen during a scheduled ECHO Preventative Maintenance window. This strategy will also include a way for Data Partners to test the new ACL structure without compromising their data's access integrity. It is likely that the old and new ACL systems will be able to coincide within ECHO to facilitate testing and an eventual cutover.

1.4.3 Provider Role

Due to the removal of the default provider context, providers may have to select a provider before performing provider related operations. This is the current behavior with PUMP for users with more than one provider role. Without the default provider context and without provider ACL exception, provider users will not be able to see restricted data without adding themselves to the proper groups.

1.5 Client Partner Impact

Clients that support rule manipulation will need to be updated to work with the new rule API. Currently PUMP is the primary ACL management client and it will be updated as part of this effort. The current rule management API on the data management service will be removed. The associated changes to group management that may be made along with this ticket have the potential to impact existing clients that manage groups. The specific details will be included at a later date after additional internal design.

2 Conceptual Approach

The general conceptual approach is based on concepts from Role Based Security Best Practices (Ferraiolo, Kuhn, & Chandramouli, 2003) and implementation concepts from Spring Security (Alex & Taylor, 2009). For the remainder of this document, the terms group and role can be used interchangeable and represent the same concept.

2.1 Overview

The proposed ACL system is based on the concept that a user is identified by a security identity (SID), which is their membership in one or more groups. These ECHO user groups are assigned permissions to objects within the ECHO system. These objects are uniquely identified by an object identity. For example, an object may be the “provider policies for provider X” or “data set ID TEST 123”.

An ACL is responsible for linking groups to an object identity and describing the permissions the group should be granted. The permissions for each object may be *Create, Read, Update, Delete and/or Order*. The grantable permissions will vary based on the domain object in question. For example, the *Provider Policies* object can be granted the *Read* or *Update* permissions, while the *Provider Audit Report* object can only be granted the *Read* permission. The *Order* permission is only used in the context of controlling the ability to order catalog items.

2.2 Terms

This section defines the terms used in the ACL system.

Term	Definition	Example
Security Identity (SID)	A Security Identity (SID) represents an identity used to link a user to a permission.	User ‘mcechini’ is identified as being a part of group ‘ECHO_Operations’
Permission	The specific permission granted to an SID. In most cases these permissions include read and write but may vary based on the object type.	Create, Read, Update, Delete, Order
Object Identity	Identifies an object in the system.	LPDAAC Provider Policies Granule Catalog Item identified by ECHO Granule ID ‘G-LARC-12345’
Access Control Level (ACL)	An access control level is the association of an object identity to a listing of permission/SID combinations.	LPDAAC Provider Policies can be read and updated by the ‘LPDAAC_Data_Mgmt’ group. Dataset ‘AMSR_L1A’ can be read by the ‘MODIS_Super_Users’ group.

2.3 System Generated Group Memberships

To simplify the configuration of ACLs for large groups of user categories, ECHO will assign all users a number of system generated group memberships. These groups will behave just like normal groups, but are maintained automatically by the ECHO system, not ECHO Operations or Providers. The system generated groups will be defined by ECHO and not modifiable. The list of system generated groups includes the following groups:

- Registered Users
- Guest Users

2.4 Granting Permissions

In the existing ACL system, any user who has been given *Provider Role* may view, update, and delete ACLs controlling access to their provider's catalog items and grant or revoke these privileges to other users. In the new ACL system, there will not be a *Provider Role* with these same privileges. Instead, there will be a DAAC 'Super Users' group which will be granted permissions to manage all ACLs for all provider object identities. This group will also be given the ability to view and order all catalog items and will be granted the ability to allow others to manage Catalog Item ACLs. Users who are given the ability to manage Catalog Item ACLs will also be able to grant other groups the ability to manage Catalog Item ACLs. An alternative approach for granting ACL management is discussed in Section 6.

(See Section 3 for more information regarding ACL object identities)

2.5 Important Concepts

A few key observations can be made from the conceptual design presented thus far:

- All ACL controlled objects are restricted by default.
- If there is no explicit permission for a user based on their group membership, the authorization is denied.
- If multiple ACLs exist for a single object identity (collection), authorization is granted to a user for a specific action if the user is in a group associated with at least one ACL that has the required permission.
- A user's group memberships, including system generated groups, are used when evaluating authorization for a user accessing an object identity. This implies that membership in any permitted group is sufficient for authorization.

3 Access Controls Overview

The information in this section gives an overview of the access control capabilities that will be available for ECHO Data Partners. This includes object identities and their grantable permissions, along with the supported Catalog Item ACL conditions. Additional object identities and their permissions available to ECHO Operators can be found in Appendix A.

3.1 “Data Partner” Object Identity Permissions

The following table lists each object identity for which a data partner will have the ability to create an ACL. Each object identity’s grantable permissions are listed along with a description of what those permissions allow. Any subset of the permissions may be used as a part of an ACL. For instance, Data Managers could be given the permission to create, read, and delete Option Definitions, but only User Services team members could be given the permission to read those same definitions.

Table 1 - "Data Partner" Object Identity Permissions

Object Identity	Grantable Permission(s)	Description
Provider Audit Report	Read	Allows the viewing of an audit report for actions associated with a specific provider.
Option Assignments	Create, Read, Delete	Allows the assignment of an option definition to one or more datasets.
Option Definitions	Create, Read, Delete	Allows the creation, viewing, and deleting of an option definition.
Option Definition Deprecation	Create	Allows the deprecation of an option definition.
Visibility Flags	Read	Allows the viewing and updating of a catalog item’s visibility flag.
Dataset Information	Read	Allows the usage of the reconciliation GetDatasetInformation() method.
Provider Holdings	Read	Allows the viewing of a provider’s holdings (dataset & granule count).
Extended Services (all types)	Create, Read, Update, Delete	Allows the creation, viewing, updating, and deletion of extended services.
Provider Orders	Read	Allows the viewing of all orders associated with a specific provider.
Provider Order Resubmission	Create	Allows the resubmission of a provider’s order.
Provider Order Acceptance	Create	Allows the acceptance of a provider’s order. Order Fulfillment Service Users (EWOC) will use this ACL.
Provider Order Rejection	Create	Allows the rejection of a provider’s order. Order Fulfillment Service Users (EWOC) will use this ACL.
Provider Order Closure	Create	Allows the closure of a provider’s order. Order Fulfillment Service Users (EWOC) will use this ACL.

Provider Order Tracking Id	Update	Allows an order to be updated with a provider tracking ID. Order Fulfillment Service Users (EWOC) will use this ACL.
Provider Information	Update	Allows provider information to be updated.
Set Provider Context	Read	Allows a user to act as a provider and perform all permitted provider actions.
Authenticator Definition	Create, Read, Delete	Allows the creation, viewing, and deletion of provider authenticators. (Not currently being used)
Provider Policies	Create, Read, Update, Delete	Allows the editing of provider policies.
User	Read	Allows the viewing, updating, and deletion of an ECHO user.

3.2 Catalog Item ACL Conditions

The following table outlines the conditions that will be supported for filtering access to view and order catalog items from a provider's holdings. Any combination of conditions may be chosen as a part of an ACL. When evaluating the ACL, each condition must be met in order for authorization to be granted. Each ACL has a required catalog item type associating it with collections, granules, or both. Unlike current functionality, a granule ACL may be created for granules within a specific collection

Table 2 - Catalog Item ACL Conditions

Condition Name	Comparator
Dataset ID	Pattern Matching and Listing
Short Name	Pattern Matching and Listing
Version ID	Pattern Matching and Listing
Granule UR	Pattern Matching and Listing
ECHO Item ID	Pattern Matching and Listing
Item Acquisition Temporal Before/After*	Intersection
Item Creation Temporal Before/After*	Intersection
Item Insert Temporal Before/After*	Intersection
Item Last Update Temporal Before/After*	Intersection
Item Acquisition Rolling Temporal Range	>, >=, <, <=
Item Creation Rolling Temporal Range	>, >=, <, <=
Item Insert Rolling Temporal Range	>, >=, <, <=
Item Last Update Rolling Temporal Range	>, >=, <, <=
Restriction Flag Value / Range	>, >=, <, <= / Inclusive Intersection

* Note: The Temporal Before/After conditions will allow for matching temporal data within a temporal range (e.g. after 1/1/08 and before 1/1/09) or outside of that range (e.g. before 1/1/08 or after 1/1/09).

4 Access Controls Examples

The following sections give working examples of how the new ACL system can be used to manage access to Data Partner object identities and catalog items.

4.1 DAAC Super Users

ECHO has recommended to have each Data Partner configure a DAAC ‘Super Users’ group wherein each member of that group will have all permissions on all provider object identities, and the additional ability to grant those permissions to other users. This will allow a DAAC super user to grant the user services team specific permissions. The ability to ‘grant’ permissions for provider object ACLs is not transferable. Even though the DAAC user services team has been granted the ability to read provider policies, they cannot grant or revoke that permission from any other users. This has been done to simplify the PUMP interface and a Data Partner’s interaction with ECHO ACLs.

4.2 ECHO Operators

ECHO Data Partners will not need to explicitly manage permissions to their provider objects or catalog items for the ECHO Operations team. The ECHO Operations team will have the ability to create ACLs permitting them to access all provider objects and all catalog items. Access to all provider objects is currently allowed by the capability of an ECHO Admin to set their context to any ECHO provider.

4.3 User Services Role

Data Partners wishing to create a distinct user services ‘role’ may do so according the following steps. The first step to creating the user services ‘role’ is to create an ECHO group. Through PUMP, the group would then be granted permissions on all provider objects according to the Data Partner’s individual needs. Once this has been accomplished membership in the group becomes the single point of modification for controlling who is permitted to perform user services activities in ECHO. The following table provides an example of what permissions could be assigned to the user services group.

Table 3 - User Services Role - Sample Permissions

Object Identity	Permission(s)
Provider Orders	Read
Provider Order Resubmission	Create
Provider Order Rejection	Create
Provider Order Closure	Create
Provider Order Tracking Id	Update
Option Assignments	Read
Option Definitions	Read
Provider Policies	Read, Update
Set Provider Context	Read
User	Read

4.4 Data Management Role

Data Partners wishing to create a distinct data manager 'role' may do so according the following steps. The first step to creating the data manager 'role' is to create an ECHO group. The group would then be granted permissions on all provider objects according to the Data Partner's individual needs. Once this has been accomplished membership in the group becomes the single point of modification for controlling who is permitted to perform data management activities in ECHO. The following table provides an example of what permissions could be assigned to the data manager group.

Table 4 - Data Management Role - Sample Permissions

Object Identity	Permissions
Provider Audit Report	Read
Option Assignments	Create, Read, Delete
Option Definitions	Create, Read, Delete
Option Definition Deprecation	Create
Visibility Flags	Read, Update
Dataset Information	Read
Provider Holdings	Read
Provider Orders	Read
Provider Information	Update
Set Provider Context	Read
Provider Policies	Create, Read, Update, Delete
SSL Certificate	Read, Update
User	Read

4.5 Catalog Item ACLs

ECHO Operations will work with each Data Partner to develop a new ACL structure to ensure that the required data access controls that currently exist are achieved by the new ACL system. Data Partners will no longer need to create restriction rules, but simply manage the permissions to view and order their catalog items. The following sections show a simple ACL example and a more complex example using an existing provider ACL structure.

4.5.1 Simple ACL Example

The following example shows a possible Catalog Item ACL configuration.

Description	Permission(s)	Item Type(s)	Condition	User/Group
Allow public view and order for non-restricted data.	View, Order	Collection, Granule	Dataset ID, Granule Restriction Flag = 0	Registered Users, Guest Users
Allow Science users to view data acquired during 2008	View	Collection, Granule	Dataset IDs, Granule Acquisition	Science_Team
Allow Restricted Users to view restricted data.	View	Collection, Granule	Dataset ID, Granule Restriction Flag > 0	Restricted_Users

4.5.2 Complex ACL Example (NSIDC)

The following example uses the currently configured ACLs for the NSIDC provider. The first table shows the ACLs that are being used to control access to NSIDC catalog items in the existing structure. The second table shows the ACLs that will be created in the new ACL structure to achieve the same access control for the affected catalog items.

Table 5 - Current NSIDC ACL Structure

Description	Rule Type	Action Type	Target Item(s)
Allow NSIDC Ops to view all collections	Permit	View	All Collections
Allow NSIDC Testers to order all collections	Permit	Order	All Collections
Allow ordering of public collections	Permit	Order	Collection Listing
Allow ordering of AMSR/ADEIS-II data to approved users	Permit	Order	Collection Listing
Allow ordering of MODIS Golden Month data to approved users	Permit	Order	Collection Listing
Allow NSIDC Testers to view all collections	Permit	View	All Collections
Allow viewing of public collections	Permit	View	Collection Listing
Allow WIST_Valids to view all collections	Permit	View	All Collections
Allow ECHO Operations to view all collections	Permit	View	All Collections
Allow viewing of MODIS Golden Month data to approved users	Permit	View	Collection Listing
Restrict viewing to all collections	Restrict	View	All Collections
Restrict ordering to all collections	Restrict	Order	All Collections

Table 6 - Proposed NSIDC ACL Structure

Description	Permission(s)	Item Type(s)	Condition	User/Group
Allow NSIDC Ops and NSIDC Testers to view all collections	View	Collection, Granule	Dataset IDs	NSIDC_Ops, NSIDC_Testers
Allow NSIDC Testers to order all collections	Order	Collection, Granule	Dataset IDs	NSIDC_Testers
Allow viewing & ordering of MODIS Golden Month data to approved users	View, Order	Collection, Granule	MODIS Dataset IDs	MODIS_Group
Allow ordering of AMSR/ADEOS-II data to approved users	Order	Collection, Granule	AMSR/ADEOS Dataset IDs	AMSR_Group
Allow viewing & ordering of public collections	View, Order	Collection, Granule	Dataset IDs	Registered Users, Guest Users

5 PUMP Interface Mockups

The following sections outline the proposed interface changes to PUMP. Due to the distinctly different nature of ACLs controlling access to provider object identities (e.g. provider policies) and ACLs controlling access to catalog items, the two ACL object types will be presented as two different options in the provider context left-hand navigation panel. These options will be labeled *Provider Object ACLs* and *Catalog Item ACLs*. The workflow for each of these navigations follows.

5.1 Viewing/Editing Provider Object ACLs

Within a 'provider context' in PUMP, when a user selects the *Provider Object ACLs* option, the interface will present with a listing of all groups associated with the provider and the system generated groups. The following figure shows how this will look in PUMP.

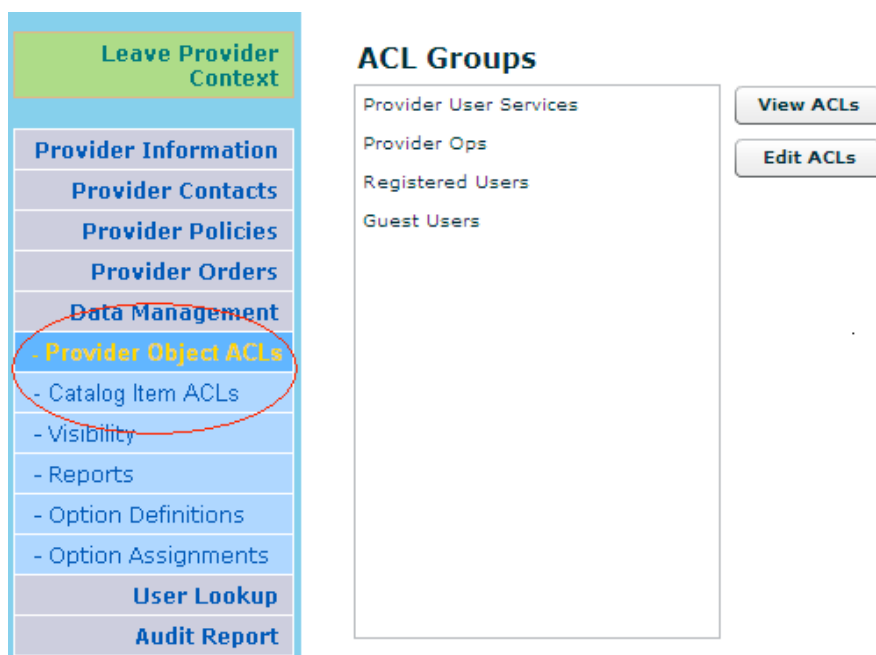


Figure 1 - Provider Object ACLs Main View

A user who has permissions to modify ACLs may select a group and edit the ACLs that are associated with that group, while an un-permissioned user will only be allowed to view the associated ACLs. After choosing to edit Provider Object ACLs for a specific group, a user will be presented with a listing of all object identities and the permissions to which the selected group has been assigned. While editing the group's assigned permissions, the user may modify the assigned permissions to each of the provider object identities. Users who are only able to view permissions will be presented with the same information, but will not be allowed to make changes. The following figure shows how the object identity permissions can be displayed in PUMP.

ACLs for Group Provider User Services

Controlled Object	Create	Read	Update	Delete
Provider Orders	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Provider Profile	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Provider Policies	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Option Assignments	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Provider Audit Report	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Option Definition Deprecation	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure 2 - Provider Object Identity ACLs

5.2 Viewing/Editing Catalog Item ACLs

Within a 'provider context' in PUMP, when a user selects the *Catalog Item ACLs* option, the interface will present a listing of all groups associated with the provider and the system generated groups. This is shown in Figure 1 above.

A user who has permissions to modify ACLs may select a group and edit the ACLs that are associated with that group, while an un-permissioned user will only be allowed to view the associated ACLs. After choosing to edit Catalog Item ACLs for a specific group, a user will be presented with a listing of all catalog item ACL descriptions (describing the conditions) and the permissions assigned to that ACL. This is shown in Figure 3, below.

ACLs for Group Provider User Services

Catalog Item ACL Description	View	Order	Edit	Delete
AMSR/ADEOS-II Collections	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>
MODIS Golden Month	<input type="checkbox"/>	<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>
All Collections for Testing	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>
All Public Collections	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>
All Invalid Granules	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>

☐ Allow catalog item ACL management

Figure 3 - Catalog Item ACLs

While editing the group's permissions, the user may modify the permissions assigned to each catalog item ACL. If members group should also have the ability to create, update, and delete catalog item ACLs, then the *Allow catalog item ACL management* checkbox should be checked. This will grant permissions to all members of the existing group to edit all existing catalog item ACLs. In addition to assigning permissions to existing ACLs, this page in PUMP allows a user to edit existing ACLs or create new catalog item ACLs. It should be noted that deleting an ACL in the context of a single group will modify that ACL for all groups to which it is assigned.

When choosing to edit or create a new ACL, the PUMP page found in the following figures will be shown. Figure 4 shows the entire page with the *Collection* filters tab selected. Figure 5 shows only the contents of the *Granule* filters tab. The *Apply ACL to:* options designate which catalog item types will be affected by this ACL.

Edit Catalog Item ACL Identifier

Name:

Apply ACL to: ☐ Collections ☒ Granules ☐ Both

Collection identification:

☐ List:


Data Set ID	Short Name	Version

☐ Pattern:


☐ Any

Filters: **Collection** Granule

Temporal Type: **ACQUISITION** ▼



Before: 

12 : 02 : 00

After: 

12 : 02 : 00

Restriction flag: ▼

0  

Note: All filters must match for the ACL to apply to a catalog item.

Save **Cancel**

Figure 4 - Edit Catalog Item ACL w/ Collection Filters

Figure 5 - Edit Catalog Item ACL - Granule Filters

6 Additional Items

As of the time when this document has been created and distributed, the following issues require resolution.

6.1 Group Management

Group management and permissions are currently based upon provider context and whether a user is a manager of that group. This functionality is inconsistent with the proposed changes in this Ops Concept. Also, there is currently no method for obtaining groups that are associated with a given provider. These issues will be addressed in a separate Ops Concept and then combined into a single document.

6.2 Granting Catalog Item Permission

In the currently proposed ACL system, controlling permissions to catalog item ACLs does not require membership in the DAAC Super Users group, mentioned previously in this document. Any user who has permissions to manage ACLs can assign or revoke that management capability to or from another user. It is therefore possible for a member of the user services group to remove the data management group's permissions to view catalog items. This is a departure from how provider objects are controlled, but has been chosen so that the ability to manage ACLs can be shared.

An alternative approach would be to remove the ability of sharing ACL management from PUMP and therefore disallow a group to add or remove another groups ACL management ability. If a Data Partner wanted to grant a group the ability to manage ACLs, then they could coordinate this with ECHO Operations. It is likely that this will not occur often outside of provider setup, but does require coordinating with ECHO Operations a group's ACL management needs change.

7 Appendix A - ECHO Operations

7.1 Terms

The following table and diagram give detailed descriptions regarding the ACL concepts that are being utilized and how they are related. For the purpose of the previous portion of this document, these terms were simplified or removed so as to not disrupt the review with technical details.

Security Identity (SID)	A Security Identity (SID) represents an identity used to link a user to a permission. The only supported SIDs in ECHO are GrantedAuthorities which map to group names.	Group Membership
Permission	The specific permission granted to an SID. In most cases these permissions include read and write but may vary based on the object type.	Create, Read, Update, Delete
Access Control Level (ACL)	An access control level is the association of an object identity to a collection of access control entries.	A user identified by id 'Test1' has the 'Read' permission on dataset 'MODIS V1'
Access Control Entry (ACE)	An association of a single SID to a Permission.	A user identified by id 'Test1' has the 'Read' permission
Object Identity	Identifies an object in the system using a combination of an object type and object identifier.	Granule Catalog Item identified by ECHO Granule ID 'G-LARC-12345'
Object Identifier	A component of the object identity that uniquely identifies an object. In the ECHO domain, object identifiers are normally represented by a GUID.	ECHO Granule ID 'G-LARC-12345'
Multi-Object Identifier	Identifies one or more objects in the system using some characteristic of the object such as the value of one or more fields in the object. A multi-object identifier can be thought of as a composite of conditional object identifiers.	All datasets with long name containing 'MODIS'.
Granted Authority	A security identity (SID) granted to a user. In the ECHO domain, a granted authority implies a group membership.	Group 'A', Registered User, Guest User

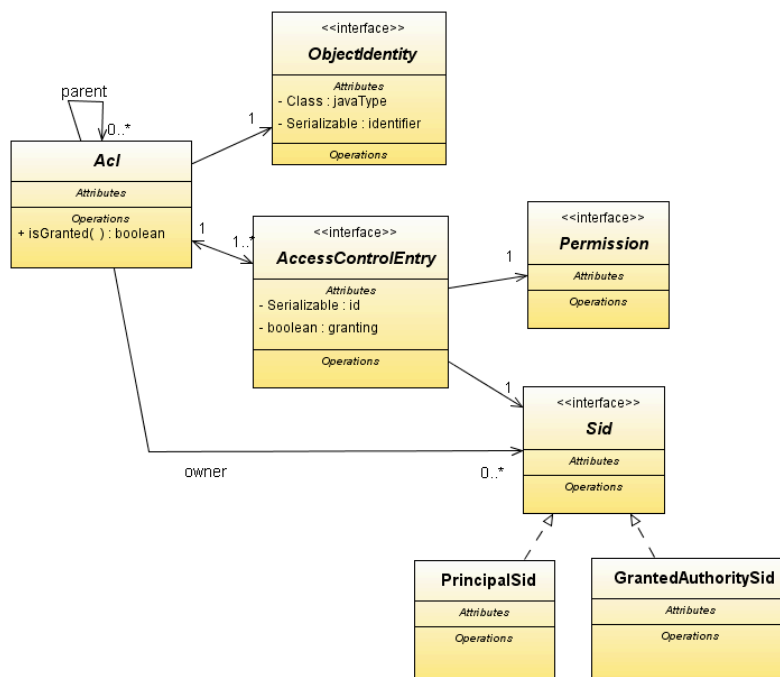


Figure 6: Object relationships in the ACL system (Borkowski, 2008)

7.2 “System” Domain Object Permissions

The following table lists each domain object for which ECHO Operations has the ability to create an ACL. Each domain object’s grantable permissions are listed along with a description of what those permissions allow.

Domain Object	Grantable Permissions	Description
System Audit Report	Read	Allows the viewing of an audit report for recorded actions.
Metric Data Point Sample	Read	Allows the generation of ECHO metrics samples.
System Initializers	Create	Allows the initial system initialization used during a first-time deployment.
Archive Records	Delete	Allows the purging of archived records.
Error Messages	Update	Allows the updating of ECHO custom error messages.
UDDI Registry Synchronization	Create	Allows the request for UDDI synchronization.
Token	Read, Delete	Allows the deletion and viewing of security token’s information.
Token Revocation	Create	Allows the revocation of a security tokens.
Extended Services Activation	Create	Allows the activation of an extended service.

Orders and Order Items	Read, Delete	Allows the viewing and deleting of all ECHO orders and order items.
Provider (inactive)	Create, Read, Delete	Create, Read, and Delete inactive providers.
Provider Activation	Create	Allows the activation of a new ECHO provider.
SSL Certificate Activation	Create	Allows the activation of a provider's SSL certificate.
Taxonomy	Create	Allows the creation of a taxonomy.
Taxonomy Entry	Create	Allows the creation of a taxonomy entry.
User Context (act as user)	Read	Allows a user to act as another user.
User	Read, Update, Delete	Allows the viewing and updating of user information and the deletion of users.

Note: This is an extension of the 'User' object listed in the provider object section, but is the same object, just with it's full permissions listing.